UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/778,076	02/07/2001	Yutaka Haga	1046.1236/JDH	4573

21171 7590 06/18/2007

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
YIGDALL, MICHAEL J	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
06/18/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

JUN 18 2007

Technology Center 2100

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/778,076
Filing Date: February 07, 2001
Appellant(s): HAGA, YUTAKA

Paul W. Bobowiec
For Appellant

EXAMINER'S ANSWER

This is in response to the substitute appeal brief filed on February 20, 2007 appealing from the Office action mailed on July 12, 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct. The examiner notes, however, that the Office action mailed on July 12, 2006 was a non-final action.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,002,872	ALEXANDER, III et al.	12-1999
6,253,338	SMOLDERS	6-2001
6,427,206	YEH et al.	7-2002

(9) Grounds of Rejection

The following ground(s) of rejection, set forth in the Office action mailed on July 12, 2006 and incorporated herein, are applicable to the appealed claims:

- Claims 8-17, 19-28 and 30-42 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,002,872 to Alexander, III et al. ("Alexander") in view of U.S. Patent No. 6,253,338 to Smolders ("Smolders") and in view of U.S. Patent No. 6,427,206 to Yeh et al. ("Yeh").

Claim 12

Alexander discloses an apparatus for collecting a profile of a subroutine included in a program (see, for example, the title and abstract), comprising:

(a) a storage unit storing a profile (see, for example, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, noting that each node is considered a storage unit);

(b) an analyzing section, when an interrupt is generated during execution of said program: obtaining a branch source address and a branch destination address from a source of said interrupt (see, for example, column 5, lines 20-32, which shows analyzing the stack frames

in response to an interrupt to identify subroutines, and column 5, lines 41-62, which shows obtaining a call or branch source address and a return or branch destination address).

Although Alexander discloses generating timer interrupts (see, for example, column 4, lines 20-23), and suggests that other interrupts may be generated instead (see, for example, column 11, lines 22-25), Alexander does not expressly disclose the limitation wherein the interrupt is generated by execution of a branch instruction, and Alexander does not expressly disclose identifying a type of said branch instruction by obtaining an instruction code from said branch source address and decoding said instruction code.

However, Smolders discloses generating an interrupt by execution of a branch instruction (see, for example, column 3, lines 58-61). Smolders further discloses an instruction flow unit that dispatches instructions to selected execution units for execution (see, for example, column 3, lines 10-13). The execution units include fixed-point execution units, load/store execution units, and floating-point execution units (see, for example, column 3, lines 13-15). The instruction flow unit cannot dispatch instructions in this manner without first interpreting or "decoding" the instruction codes, so as to determine the appropriate execution unit. Furthermore, it is understood in the art that such instruction decoding is an integral part of the instruction cycle.

Thus, Smolders discloses a system for collecting a trace of a program (see, for example, the title and abstract), wherein an interrupt is generated by execution of a branch instruction, and wherein the type of the branch instruction is inherently identified by obtaining an instruction code and decoding the instruction code. The system enables tracing without introducing any overhead and without modifying the code (see, for example, column 1, lines 64-67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the profiling system of Alexander with the features taught by Smolders and to substitute the timer interrupt of Alexander with the branch interrupt of Smolders, as suggested by Alexander, so as to obviate any overhead and modifications to the code.

Moreover, Yeh expressly discloses obtaining an instruction code and decoding the instruction code (see, for example, column 3, lines 45-49), and identifying the type branch, such as whether the branch is a calling instruction or a return instruction (see, for example, column 3, lines 63-65), so as to collect branch predictions and provide prediction hints (see, for example, column 3, lines 52-60). The branch prediction system of Yeh enhances profiling (see, for example, column 2, lines 1-13) to enable speculative execution of instructions (see, for example, column 1, lines 29-40).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the profiling system of Alexander with the features taught by Yeh and to identify the type of branch, as taught by Yeh, so as to collect branch predictions and enhance the collection of profiles for purposes of speculative execution.

Alexander in view of Smolders in view of Yeh further discloses:

(c) a collecting section: obtaining said branch source address, said branch destination address, and an identified result from said analyzing section when the identified instruction is a calling instruction or a return instruction of said subroutine (see, for example, Alexander, column 5, lines 41-62, which shows obtaining a call or branch source address and a return or branch destination address for a subroutine); and when said identified result is said calling instruction, storing said branch destination address as a subroutine address corresponding to said calling

instruction and a calling time of said subroutine corresponding to said calling instruction in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and a base time), when said identified result is said return instruction, obtaining a return time of said subroutine corresponding to said return instruction, calculating a execution time of said subroutine based on said obtained return time and said calling time, and storing a cumulative value of said execution time as said profile in correspondence with said branch destination address in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including an execution time and a cumulative time, inherently calculated based on the calling time and the return time, and see, for example, FIG. 4A, which shows timestamps for entering and returning from subroutines), and when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated (see, for example, Yeh, step 310 in FIG. 3 and column 8, lines 42-48, which shows not collecting history information when the branch is not a calling instruction or a return instruction, and see, for example, Smolders, steps 46 and 64 in FIG. 3 and column 5, lines 43-52, which shows terminating the interrupt when trace information is not to be collected).

Claim 8

The rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein a plurality of storage units respectively corresponding to a plurality of executors of said subroutine are prepared (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a plurality of nodes or storage units); and

said collecting section specifies said executor of said subroutine and stores said profile of said subroutine corresponding to said specified executor in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles; note the parent node pointers, which specify the executors of subroutines).

Claim 9

The rejection of claim 8 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section individually stores profiles of a plurality of subroutines corresponding to a specified executor in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, noting that a path in the tree represents the subroutines executed by a specific executor).

Claim 10

The rejection of claim 9 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section individually stores a first profile of a subroutine called by a main routine, and a second profile of the subroutine called by another subroutine, in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, and FIG. 10 and column 8, lines 24-40, which shows individual profiles of subroutines organized based on the calling routine, and see, for example, FIG. 8, which shows a subroutine Y called by both the main routine and by a second routine X).

Claim 11

The rejection of claim 10 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section stores said second profile and calling relationship information relating to said second profile, said calling relationship information indicating a relationship between said other subroutine and said called subroutine, in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including calling relationship information).

Claim 13

The rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section stores times of calling of said subroutine corresponding to said branch destination address as said profile in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and a base time, and see, for example, FIG. 4A, which shows timestamps for entering, i.e. times of calling, and returning from subroutines).

Claim 14

The rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section obtains an overhead of said subroutine as said profile and stores said overhead in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the

profiles, including a time consumed by a thread executing a subroutine, which is considered a measure of overhead).

Claim 15

The rejection of claim 13 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section, when said identified result is said calling instruction, stores an identifier of an executor of said subroutine corresponding to said calling instruction and said branch destination address in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address, and note the parent node pointers, which identify the executors of subroutines).

Claim 16

The rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section, when said identified result is said calling instruction and said branch source address and said branch destination address are addresses of said subroutines, stores said branch source address and branch destination address as calling relationship information indicating a callings source subroutine and a calling destination subroutine is said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and calling relationship information), and stores at least one of the cumulative execution time and the times of calling in said calling destination subroutine in the call source subroutine, as said profile corresponding to said calling relationship information, in said storage unit (see, for

example, Alexander, column 6, lines 37-53, which shows that the profile comprises both a base time and a cumulative execution time).

Claim 17

The rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses a setting section setting an execution environment of a source of said interrupt so as to generate said interrupt when said branch instruction is executed during the execution of said program (see, for example, Smolders, column 3, lines 58-61, which shows setting the execution environment to generate a trace interrupt after every branch instruction).

Claim 23

The computer readable medium recited in the claim is analogous to the apparatus recited in claim 12 (see the rejection of claim 12 above, and see, for example, Alexander, column 11, lines 6-17, which shows a computer readable medium).

Claims 19-22 and 24-28

The limitations recited in the claims are analogous to those of claims 8-11 and 13-17, respectively (see the rejection of claims 8-11 and 13-17 above).

Claim 34

The method recited in the claim is analogous to the apparatus recited in claim 12 (see the rejection of claim 12 above).

Claims 30-33 and 35-39

The limitations recited in the claims are analogous to those of claims 8-11 and 13-17, respectively (see the rejection of claims 8-11 and 13-17 above).

Claim 40

The rejection of claim 13 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein the collecting section generates a control table corresponding to each executor of the subroutine on the storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles; note that a path in the tree represents the subroutines executed by a specific executor, and see, for example, FIG. 9 and column 8, lines 24-33, which shows the data structure in the form of a table),

wherein the control table includes an executor managing table, a subroutine managing table, and a calling managing table (see, for example, Alexander, column 6, lines 54-61, which shows that the data structure may include other pointers and tables to aid in subsequent analysis).

wherein the executor managing table stores an identifier of the executor and a pointer to assign the subroutine managing table (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows parent node pointers that identify the executors of subroutines),

wherein the subroutine managing table is generated for every subroutine executed by the executor, the subroutine managing table storing a subroutine address, times of calling of the subroutine, a cumulative execution time of the subroutine, the last called time of the subroutine, and a pointer to assign the calling managing table (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows a subroutine address, a base time and a cumulative time, and

see, for example, FIG. 4A, which shows timestamps for entering and returning from subroutines), and

wherein the calling managing table is generated for every subroutine called by the subroutine, the calling managing table storing a branch source address as a calling subroutine address, a branch destination address as a called subroutine address, times of calling of the called subroutine, a cumulative execution time of the called subroutine, the last called time of the called subroutine, and a pointer to specify the subroutine managing table managing the calling subroutine (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows a subroutine address, a base time and a cumulative time, and FIG. 4A, which shows timestamps for entering and returning from subroutines, and see, for example, column 5, lines 41-62, which shows a call or branch source address and a return or branch destination address for a subroutine).

Although Alexander in view of Smolders in view of Yeh does not expressly disclose the recited table names and pointers, Alexander in view of Smolders in view of Yeh discloses that the data structure may include such pointers and tables to aid in subsequent analysis (see, for example, Alexander, column 6, lines 54-61). Alexander in view of Smolders in view of Yeh further discloses the recited information stored in the data structure, as presented above.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the data structure of Alexander in view of Smolders in view of Yeh with other pointers and tables, as suggested by Alexander, including an executor managing table, a subroutine managing table, and a calling managing table, each with corresponding pointers, for the purpose of facilitating subsequent analysis of the profiles.

Claims 41 and 42

The limitations recited in the claims are analogous to those of claim 40 (see the rejection of claim 40 above).

(10) Response to Argument

a. Claim 12 (brief, page 11)

Appellant states, generally, that the combination of the Alexander, Smolders and Yeh references does not discuss "identifying a type of said branch instruction by obtaining an instruction code from said branch source address and decoding said instruction code ... and when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated" (brief, page 12).

Specifically, Appellant contends that the examiner's statement that it would have been obvious to one of ordinary skill in the art to substitute the timer interrupt of Alexander with the branch interrupt of Smolders is conclusory and does not properly support an establishment of *prima facie* obviousness (brief, page 13).

However, the examiner respectfully submits that a conclusion of obviousness is indeed supported. The examiner acknowledges that Alexander describes the timer interrupt as one example of a periodic event:

> A sample is obtained each time a periodic event, such as, for example, a timer interrupt occurs. (Column 5, lines 37-38.)

Nonetheless, as Appellant concedes (brief, page 13), Alexander expressly suggests the use of other kinds of interrupts:

> Many modifications and variations will be apparent to those of ordinary skill in the art. For example, although the depicted examples employ timer interrupts, other interrupts may be used to trigger the described sampling mechanism. (Column 11, lines 21-25.)

Therefore, based on Alexander's teachings alone, it would have been obvious to one of ordinary skill in the art to use another kind of interrupt to trigger the sampling mechanism. Smolders teaches the use of one such kind of interrupt to trigger a comparable sampling mechanism (see, for example, the abstract).

Specifically, Smolders teaches a trace interrupt that is generated after every branch instruction (see, for example, column 58-61). The trace interrupt is thus considered a "branch interrupt" in the Office action. Smolders' teachings enable "tracing hardware counters by way of an interruption without introducing any overhead or modifying the code" (see, for example, column 1, lines 64-67). Therefore, it would have been obvious to one of ordinary skill in the art to use a branch interrupt to trigger the sampling mechanism of Alexander. As Smolders suggests, this would enable sampling without introducing any overhead and without modifying the code of the program.

Even if one construes Alexander such that the sampling mechanism is strictly limited to interrupts that are periodic events, a reasonable interpretation of a "periodic event" is an event that repeats or is recurring. The trace interrupt of Smolders is a recurring event. While Appellant concludes that Smolders does not describe the trace interrupt as a periodic event (brief, page 13), the cited passage clearly states that the interrupt is generated "after each branch" (see, for example, column 3, line 58 to column 4, line 6). Thus, one of ordinary skill in the art would recognize that the trace interrupt is a recurring event.

Moreover, Alexander teaches an embodiment that uses a recurring event other than the timer interrupt to trigger the sampling mechanism:

> Instead of employing a timer interrupt, a page fault interrupt may be used as a signal to gather data from the stack. In the depicted example, a page fault is forced each time a memory access occurs within the data processing system. A page fault also may be selectively triggered, depending on processing resources. (Column 8, lines 51-56.)

Appellant's statements (brief, page 14) notwithstanding, the page fault interrupt of Alexander is a recurring event in the same sense that the trace interrupt of Smolders is a recurring event. While another interpretation of a "periodic event" is an event that occurs at a fixed frequency or interval, one of ordinary skill in the art would recognize that the page fault interrupts of Alexander are not necessarily generated at a fixed frequency or interval. Thus, a conclusion that the sampling mechanism of Alexander is strictly limited to interrupts that are fixed-frequency events is not supported. The discussion of the page fault interrupt in Alexander is, at minimum, evidence in favor of a reasonable expectation of success in substituting the timer interrupt with another recurring event such as Smolders' trace interrupt.

The examiner respectfully submits that Appellant's arguments regarding the return address in Smolders (brief, page 14) are not directed to any features recited in the claims, nor are they directed to any rejections set forth in the Office action.

Rather, these arguments are presumably directed to the examiner's earlier reasoning that Smolders inherently teaches "identifying a type of said branch instruction" (see the Office action mailed on November 19, 2004). Applicant's amendment filed on February 22, 2005 necessitated new grounds of rejection involving the Yeh reference, and the inherency argument was rendered moot (see the Office action mailed on June 24, 2005).

Nonetheless, the examiner does not agree with Appellant's conclusion that Smolders does not provide a return address (brief, page 14). Smolders states:

> Thereafter, the counter level tracing tool 31 saves the address of the beginning of the next basic block of code, which is the address where the interruption came from as shown in step 34. (Column 4, lines 31-34.)

Smolders further illustrates in step 34, "Next Block = Return Address" (FIG. 3). In other words, Smolders teaches saving the address of the beginning of the next basic block of code (i.e., "Next Block" in step 34 of FIG. 3) as a return address (i.e., "Return Address" in step 34 of FIG. 3). While the address of the beginning of the next basic block of code is indeed "the address where the interruption came from," as Appellant emphasizes (brief, page 14), it is also the address that the counter level tracing tool 31 saves as a return address. One of ordinary skill in the art would recognize that the interrupt returns to the address where the interruption came from.

Again, however, these arguments are moot. As set forth in the Office action, Yeh teaches identifying the type of branch instruction and whether the branch instruction is a calling instruction or a return instruction (see, for example, column 3, lines 63-65).

b. Claim 23 (brief, page 15)

Here, Appellant refers to the arguments addressed above.

c. Claim 34 (brief, page 15)

Here, Appellant refers to the arguments addressed above.

d. Claim 40 (brief, page 16)

Appellant states that none of the cited art discusses "an executor managing table, a subroutine managing table, and a calling managing table" (brief, page 16).

However, the examiner notes that the test for obviousness is not that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981).

The claimed subject matter amounts to a data structure that includes arbitrarily named tables. Alexander's data structure is arranged in the form of a tree (see, for example, FIG. 5). While Alexander does not expressly disclose the arbitrary table names recited in the claim, such names do not represent functional distinctions. Indeed, the data stored in Alexander's data structure is analogous to the data stored in the claimed data structure (see, for example, column 6, lines 37-53), and thus both Alexander's data structure and the claimed data structure provide analogous information to those of ordinary skill in the art.

One of ordinary skill in the art would recognize that different arrangements of data often represent the same information. For example, Alexander illustrates a table in FIG. 4A, and states, "The performance data associated with the detailed trace and call stack data in FIG. 4 can be represented in a tree format" (column 6, lines 32-34). Thus, the tree structure illustrated in FIG. 5 represents the same information as the table of FIG. 4A. Alexander's FIG. 9 shows another table derived from the same data.

In response to the examiner's reasoning, Appellant contends that the pointers recited in the claim establish "specific relationships" that do render the claimed data structure functionally distinct from Alexander's data structure (brief, pages 16-17).

However, the examiner respectfully submits that one of ordinary skill in the art would recognize that such pointers are merely necessary elements of the data structure. While

Appellant emphasizes that one pointer is "to assign" a table and that another pointer is "to specify" a table (brief, pages 16-17), a reasonable interpretation is that functionally, each pointer merely points to a table. One of ordinary skill in the art would recognize that such pointers are necessary to allow the elements of the data structure to be located, accessed, traversed and so on. The elements of Alexander's data structure are similarly linked with pointers:

> In the depicted example, each node, nodes 502-508, contains an address (addr), a base time (BASE), cumulative time (CUM) and parent and children pointers. (Column 6, lines 43-46).
>
> In the depicted example, pointers are included for each node. One pointer is a parent pointer, a pointer to the node's parent. Each node also contains a pointer to each child of the node. (Column 6, lines 50-53).

Furthermore, Alexander expressly suggests other implementations of the data structure, including the use of tables and other pointers:

> Those of ordinary skill [in the] art will appreciate that tree structure 500 may be implemented in a variety of ways and that many different types of statistics may be maintained at the nodes other than those in the depicted example. In addition, other pointers may be stored within the nodes to further aid subsequent analysis. Further, other structural elements, such as tables for properties of the routine, such as, for example, the name of the routine, also may be stored within a node. (Column 6, lines 54-61.)

Therefore, it would have been obvious to one of ordinary skill in the art to implement the data structure of Alexander in a form suitable for subsequent analysis. That is to say, Alexander suggests that one of ordinary skill in the art might choose to implement the data structure in any form suitable for the subsequent analysis that he or she intends to perform.

Again, as noted above, the information that Alexander's data structure provides is analogous to the information that the claimed data structure provides. The arbitrary table names in the claimed data structure do not represent functional distinctions. Alexander suggests

including such tables in the data structure, and teaches that the data structure includes the necessary pointers. Therefore, the examiner respectfully submits that the record establishes a *prima facie* case of obviousness, and that the prior art would have suggested the claimed subject matter to those of ordinary skill in the art. Appellant does not provide any evidence of new or unexpected results owing to the arrangement of data recited in the claim.

e. Claims 8, 9, 10, 11, 13, 14, 15, 16 and 17 (brief, page 17)

Here, Appellant refers to the arguments addressed above.

f. Claims 19, 20, 21, 22, 24, 25, 26, 27, 28 and 41 (brief, page 18)

Here, Appellant refers to the arguments addressed above.

g. Claims 30, 31, 32, 33, 35, 36, 37, 38, 39 and 42 (brief, page 19)

Here, Appellant refers to the arguments addressed above.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Michael J. Yigdall
Examiner MY
Art Unit 2192

Conferees:

Tuan Q. Dam, SPE 2192

TUAN DAM
SUPERVISORY PATENT EXAMINER

Eddie C. Lee, SPE 2100

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER